

Tests of the GenWQE/PCIe Accelerator for the European XFEL

IT & Data Management Group
European XFEL
23.11.2016

Test data files

- Raw data files from LCLS detectors:
 - eXtended Tagged Container - (.xtc) format
 - **XTC streams** from LCLS experiments
- “Data files” / diffraction pattern - (**.cxi**) format
 - HDF5, NeXus-inspired and ~compatible
- Data available from **<http://cxidb.org>**
 - kindly provided by Dr. Adrian Mancuso (SPB/SFX)

LCLS: SLAC Linac Coherent Light Source

CXIDB: Coherent X-ray Imaging Data Bank

Types of experimental data

- Sequential crystallography: [idb22](#)
 - beamline CXI @LCLS
- Not-so-weakly scattering: [idb30](#)
 - beamline AMO @LCLS
- Weakly scattering – TODO

Compression rates with FPGA

- FPGA = GenWQE/PCIe GZIP Accelerator @p8.desy.de



- Comparison criteria:

space saving = $1 - 1/\text{comp_ratio}$

- $\text{comp_ratio} = \text{uncompressed_size} / \text{compressed_size}$

data compression rate (speed)

- I/O from/to 3 alternatives: disk / memory / null
- Disk: home@p8.desy.de

Compression rates with FPGA

- Sequential crystallography, idb22, ~175GB of 16 TB
 - Space saving [41-51%], depends on run #
 - Data rate (single thread):
 - Disk: 0.95 GB/s **RAM: 1.05 GB/s** RAM/null O: 1.12 GB/s
- Not-so-weakly scattering, idb30, ~210 GB of 4 TB
 - Space saving [32-42%], depends on run #
 - Data rate (single thread):
 - Disk: 0.85 GB/s **RAM: 0.89 GB/s** RAM/null O: 1.00 GB/s

Comparison against software

- Speed / data rate
 - FPGA: ~1GB/s. ~100x faster than software:
 - id22: [8.6, 8.8] MB/s
 - id30: [8.7, 9.9] MB/s
- Storage saved:
 - FPGA: [32, 51]% raw data storage save
 - Software could save more, [41, 57]%.
 - id22: [48, 57]% => ~[12,17]% (relative) higher space saving
 - id30: [41,50]% => ~[17,27]% (relative) higher space saving

Other software implementations (gzip, custom 'gzip') can be 6-7% faster
+ on different machines (exflpclXXnY), up to [19, 21]% faster.

Other remarks

- Acceleration works well with our technologies:
 - HDF5 format compression
 - Karabo framework
 - Karabo HDF5 API
 - Devices
- Multi-thread/process:
 - Could do 1.7 GB/s (2 threads)
 - Recommended? Stable? How does it work?

Headlines

- Work in progress
 - Format XTC / HDF5
 - Disk I/O setup: GPFS
- Storage saving:
 - **Raw data: ~32-51%** Other files: potential $\geq 85\%$?
- Compression data rate:
 - **FPGA rates “close to” 1 GB/s.**
 - Faster than software by x[93, 128] (RAM I/O)

Qs

- Ideal rate of 1.7-1.8 GB/s?
 - Buffer sizes - 1.4 GB/s single threaded?
 - Multi-process/threading? - 1.7 GB/s - *stable*?
- GenWQE -- CAPI